Distributed TimeScaleDB (postgres)

Helpful Documentations

TimeScaleDB install guide linux	https://www.tigerdata.com/docs/self-hosted/latest/install/installation-linux
Potgres Replication Documentation	https://www.postgresql.org/docs/current/high-availability.html
Replica Setup steps - Claude	https://claude.ai/share/7fe520d6-5fbf-4a75-9dec-a427e667ba3d

Setup Postgres + TimeScaleDB on Azure VM

1. Install Postgres packages

```
sudo apt install gnupg postgresql-common apt-transport-https lsb-release wget
```

2. Run the postgres package setup script

```
sudo /usr/share/postgresql-common/pgdg/apt.postgresql.org.sh
```

3. Add the TimescaleDB package

```
echo "deb https://packagecloud.io/timescale/timescaledb/ubuntu/ $(lsb_release -c -s) main" | sudo tee /etc/apt/sources.list.d/timescaledb.list
```

4. Install the TimescaleDB GPG key

```
wget --quiet -O - https://packagecloud.io/timescale/timescaledb/gpgkey | su
do gpg --dearmor -o /etc/apt/trusted.gpg.d/timescaledb.gpg
```

5. Update your local repository list

```
sudo apt update
```

6. **Install TimescaleDB**

```
sudo apt install timescaledb-2-postgresql-17 postgresql-client-17
```

7. Tune your Postgres instance for TimescaleDB

```
sudo timescaledb-tune

// Just keep selected 'Y' for all the steps
```

8. Restart postgres

```
sudo systemctl restart postgresql
```

9. Log in to Postgres as `postgres`

```
sudo -u postgres psql
```

10. Update the password

```
\password postgres

// Just set it as 'password'
```

Setting up replicas

Preq

Make sure you have all you VMs up and running - primary and all the secondary and have postgres with timescaledb installed

Primary Server Setup

1. Edit postgresql.conf

```
listen_addresses = '*'

wal_level = replica
max_wal_senders = 3  # At least number of standbys + 1
wal_keep_size = 1GB  # Adjust based on your needs
hot_standby = on
```

2. Create a replication user

```
CREATE ROLE replicator WITH REPLICATION LOGIN PASSWORD 'your_secure
_password';
```

3. Edit pg_hba.conf to allow replication connections from standby servers

```
# TYPE  DATABASE        USER            ADDRESS                 METHOD
# For replication
host    replication     replicator      standby1_ip/32          md5
host    replication     replicator      standby2_ip/32          md5

# For regular database connections (optional but useful)
host    all             all             172.16.0.0/24           scram-sha-256
```

4. Restart Postgres

```
sudo systemctl restart postgresql
```

Secondar Server Setup

1. Stop PostgreSQL if it's running:

```
sudo systemctl stop postgresql
```

2. Empty Data directory

```
# Back up the directory (optional, for safety)
sudo mv /var/lib/postgresql/17/main /var/lib/postgresql/17/main.backup

# Create empty directory with correct permissions
sudo mkdir -p /var/lib/postgresql/17/main
sudo chown postgres:postgres /var/lib/postgresql/17/main
sudo chmod 700 /var/lib/postgresql/17/main
```

3. Run pg_basebackup:

```
sudo -u postgres pg_basebackup -h 172.16.0.4 -D /var/lib/postgresql/17/main
-U replicator -P -v -R -X stream -C -S standby1_slot

# Remove -C flag if the standby1_slot is already created or delete the slot and
rerun the above command
# command to delete the slot (run on primary)
# SELECT pg_drop_replication_slot('standby1_slot');
```

4. Edit postgresql.conf

```
sudo vi /etc/postgresql/17/main/postgresql.conf

# Change this in the config file
hot_standby = on
```

5. Start Postgres

```
sudo systemctl start postgresql
```

Verification

Primary

```
SELECT * FROM pg_stat_replication;

# This should print all the standby server
```

Secondary

```
SELECT pg_is_in_recovery();

# this should print true
```

Add TimeScaleDB extension to the database

1. Connect to a database on your Postgres instance

```
psql -d "postgres://<username>:<password>@<host>:<port>/<database-name>"
```

2. Add TimescaleDB to the database

```
CREATE EXTENSION IF NOT EXISTS timescaledb;
```

3. Check that TimescaleDB is installed

```
\dx
```